

Integration of SCORM packages into web games

Enrique Barra, Aldo Gordillo, Daniel Gallego and Juan Quemada

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

Avenida Complutense 30, 28040, Madrid, Spain

{ebarra, agordillo, dgallego, jqumada}@dit.upm.es

Abstract— This paper presents a model that enables the integration of SCORM packages into web games. It is based on the fact that SCORM packages are prepared to be integrated into Learning Management Systems and to communicate with them. Hence in a similar way they can also be integrated into web games. The application of this model results in the linkage between the Learning Objects inside the package and specific actions or conditions in the game. The educational content will be shown to the players when they perform these actions or the conditions are met. For example, when they need a special weapon they will have to consume the Learning Object to get it. Based on this model we have developed an open source web platform which main aim is to facilitate teachers the creation of educational games. They can select existing SCORM packages or upload their own ones and then select a game template in which the Learning Objects will be integrated. The resulting educational game will be available online. Details about the model and the developed platform are explained in this paper. Also links to the platform and an example of a generated game are provided.

Keywords— SCORM; Game-Based Learning; Learning Objects

I. INTRODUCTION

Many studies show that pupils are more motivated to learn with educational games [1], [2]. They feel more engagement and learn easier. This is mainly because they have grown up with computer games, they feel comfortable using them [3] and even more with the increasing popularity of video games and the spreading of handheld devices such as smartphones and tablets.

Educational video games can be really improved with the introduction of new technologies. But the development costs are very high (even higher if we consider the opportunity costs of developing another kind of video game) and teachers or educational institutions usually cannot afford it. Therefore, teachers usually have to resort to existing games. However, most of them are focused on specific topics and cannot be reused in other academic subjects. The type of educational content used in the game (e.g. video, images or audios) or the content's language are important parts of the game that need to be adequate for students bearing in mind their context (e.g. age, level, etc.).

On the other hand e-Learning standards such as SCORM (Sharable Content Object Reference Model) [4] are widely used nowadays in Learning Management Systems (LMS) and Learning Object Repositories (LOR). They offer a good possibility for teachers to easily manage their contents, reuse them or export them from one system to another. Teachers can

also create their own resources in these standard formats using different e-Learning authoring tools.

The research presented in this paper aims to build a bridge between both worlds to create adaptive educational video games. Teachers will be able to customize the learning contents inside the games and game developers will be able to create adaptive educational video games in an easier way and following a standard for the integration of the learning contents and the communication with them. Hence the intended audience of this research is the teaching community and the game developers' community.

This research is based on the possibilities that SCORM standard offers. SCORM packages can be integrated into web games the same way they are integrated into a LMS. The Learning Objects (LOs) or activities inside the SCORM package can be shown to the players when certain conditions are met or when they do specific actions. For example, when they need a special weapon they would have to consume a LO to get it. The LOs can communicate with the game the same way that they do with a LMS. For example, a quiz would assess that the pupil have passed it and so he can get an extra life in the game.

The rest of the paper is organized as follows. The next section reviews the state of the art in educational games and the SCORM standard. Section 3 introduces the possibilities of integrating SCORM packages into web games. Section 4 presents the model including its architecture and interface. Section 5 describes the SGAME platform that we have developed following this model. Finally section 6 presents some conclusions and future work.

II. RELATED WORK

According to [5] Game-Based Learning can be defined as “the use of a computer game-based approach to deliver, support, and enhance teaching, learning, assessment, and evaluation”. It is a growing trend in recent years. As educational games have potential to improve students' motivation [6], promote self-directed learning [7] and develop social and cognitive skills [8], [9]. In [10] we can find a broad literature review of computer games and serious games, where they identify and review the impact and outcomes of 129 papers.

The potential for educators to become involved in the development of educational games is substantial [9]. Interesting initiatives are providing frameworks to develop adaptive educational games, where teachers can create and

evaluate their own games and simulations from scratch. A good example of this can be the game authoring tools provided by e-UCM, a research group from Complutense University of Madrid with several related projects and research lines like Game-Tel [11] and e-Training DS [12]. There are other software tools available that make it easier to create computer games, replacing part of the programming by mechanisms in which games are constructed from simple building blocks. Examples are StageCast [13] that is particularly aimed at young kids, the products by ClickTeam [14] or Game Maker [15].

Other initiative to involve educations in the authoring of games is to make it even easier for them introducing game templates. This concept comes from the non-digital world with the *Frame Games* described back in 1980 by [16]. Frame Games are board games which are designed so that they can be filled with arbitrary educational content (for example by putting it on blank playing cards). As an analogy in the digital world, game templates are a certain type of gameplay which can be configured and filled with content by an author [17].

On the other hand learning standards are widely used to package and interact with LOs. One of the most used is SCORM. This standard was created by ADL (Advance Distributed Learning) from the US Department of Defense to promote interoperability, reusability and durability of digital educational contents. SCORM arose as an application profile that integrated several existing standards and specifications from the e-Learning field to simplify their implementation in systems and tools. It acquired the category of technical report by ISO (International Organization for Standardization) in 2009 (ISO/IEC TR 29163).

SCORM is intended to import and export LOs to and from LMSs and LORs. But as it offers a run time environment to communicate with these LOs it can be used in other contexts. A good example of this kind of use is described in [18].

III. SCORM AND GAMES

As an introduction to the model we present in this section a summary of the SCORM standard and an analysis of the possibilities of integration in games.

A. Inside SCORM

SCORM standard has been introduced in the related work section, but now we will summarize the standard to understand the possibilities it offers for the integration of the SCORM packages in other scenarios different from a LMS. SCORM 2004 4th edition is composed of three technical documents that describe different aspects and provide guidelines for the creation and use of these SCORM packages.

- *Content Aggregation Model*: Specifies the structure and format of the file known as manifest and indicates how to tag or label the content (based on IEEE Learning Object Metadata, LOM [19]) and how to package the Learning Objects (with IMS Content Packaging). The SCORM package is a zip file containing all the source

files with the learning resources and the manifest file. A resource can be a SCO (Shareable Content Object) or an asset. A SCO is a content object that will use the SCORM API (Application Programming Interface) to interact with the Run-Time Environment when it is launched and while it is running. An asset is a content object that will not use the SCORM API but that can still be used for an activity. For example, it might be a text document or an image.

- *Run-Time Environment*: Defines how the contents have to be executed, the data model for SCOs and the way of doing this communication (IEEE 1484.11.2). SCORM specifies both an IEEE standard communication data model (IEEE 1484.11.1 known as CMI model) and a custom data model. The CMI data model contains several categories of data, among others:
 - Completion and success status.
 - Score.
 - Data about various types of interactions and their status including learner responses.
 - Entry and exit status, used to determine how the SCO was launched and how and why the SCO is being terminated.
- *Sequencing and Navigation*: Defines the order in which the content will be presented to the pupil together with the navigation options that are offered. It is based on IMS Simple & Sequencing to define flow conditions in the activities.

B. Possibilities of integration

According to the Content Aggregation Model a SCORM package is a zip file containing all the resources (SCOs and assets) inside together with a manifest file. All SCOs and assets inside the SCORM package can be integrated into a website (such as a LMS) so they can be integrated into a web game. In fact there is a required *type* attribute into the resource element of the manifest file to indicate the resource type but that is always set to *webcontent* by recommendation of the SCORM standard.

Although there is no technological constraint to integrate these resources into web games we can find a huge limitation with the resource learning time. Due to the nature of games they cannot be interrupted for a long time. For example, although a one hour video can be a very good learning resource it should not be integrated into a game.

Fortunately LOM specifies an attribute for this called “typical learning time”. LOM also has other attributes that can be useful to determine if a resource should be integrated or not, those are “duration” (to indicate full duration for videos, audios or animations), “typical age range” and “difficulty”. But if these parameters are not specified in the manifest file the user will be the one in charge of deciding if a resource should be integrated or not.

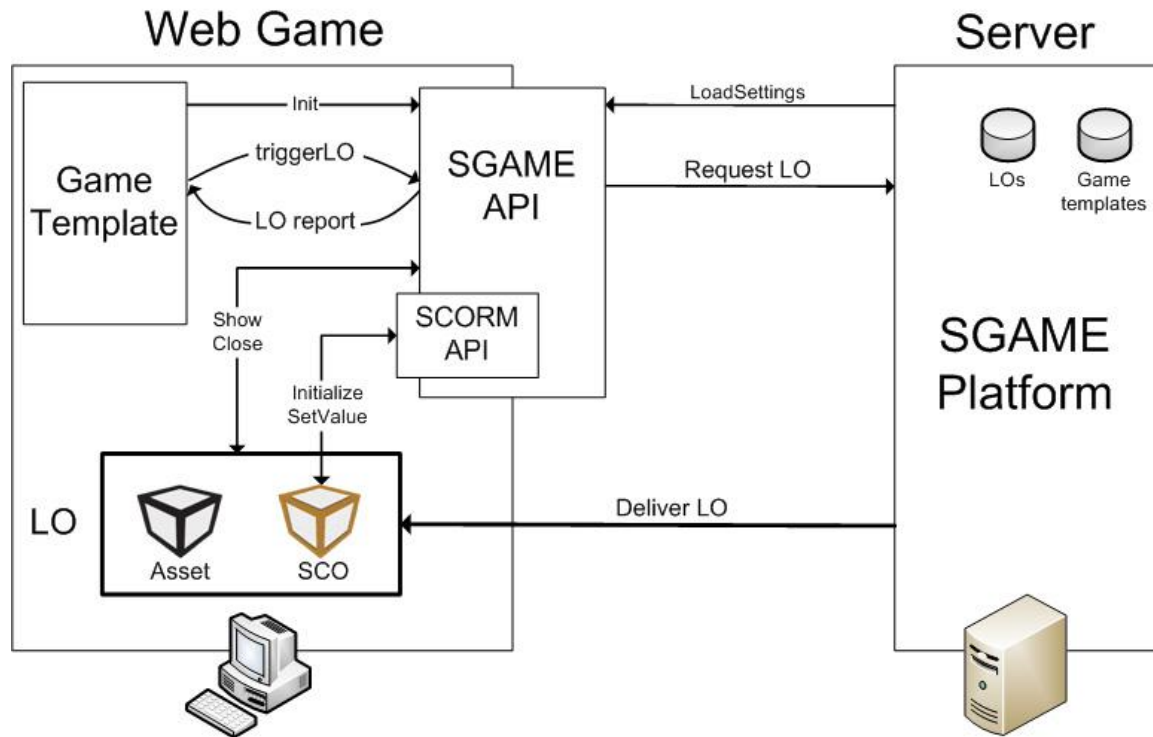


Fig. 1. SGAME Architecture

SCOs communicate their status to the LMS using the Runtime Environment, i.e. the API that SCORM defines for this communication. This status can be a score, a success or a completion status among others. By the integration of this API into the game it will be able to get information about the interaction of the player with the learning activity or LO. For example, if the player answers right to a quiz the SCO will communicate the success status through the API. The game gets this information assessing that the learning activity has been passed and the game is resumed giving the player a reward like an extra life or a better weapon.

But this interface is only used by SCOs. Assets on the other hand do not communicate any status to the LMS so they also will not do it to the game. The game can show the asset but when the player closes it the only information that can be obtained from the interaction is the time period that the asset has been displayed.

Finally although for an LMS it can be very useful, the Sequencing and Navigation part is not useful for the integration into games. The game presents the LOs to the player when the conditions are met or the actions are performed, not in a specific strict order.

IV. SGAME MODEL

In order to allow the integration of SCORM packages into web games we have defined a model called SGAME, which name is the result of the contraction of the words "SCORM" and "GAME".

This model defines the architecture and the API that allow the integration. Both will be explained in this section together with the conventions that a game developer would have to follow in order to create a game compliant with this model.

A. SGAME Architecture

Fig. 1 shows the detailed architecture and interactions between the different components of the model that are carried out for integrating SCORM packages into web games. In the figure we can see the different pieces that take part in the process.

We have already introduced SCORM related concepts but before explaining this figure we should introduce the rest of them.

- A *Web Game* (or game instance) is a game generated with the SGAME Platform composed by a game template and a set of LOs. In a specific moment this Web Game can be showing the game to the player or a LO to be consumed.
- *Game Templates* are full games with specific events that may call external LOs to be shown when they are triggered. They use the SGAME API to request them.

Taking a quick look to Fig 1 we can see two main parts, the SGAME Platform and a Web Game. The SGAME Platform is in the cloud so it can be accessed from anywhere and at any time. It stores the SCORM packages, the game templates and the generated games. This server has a very simple functionality, extract the LOs from the SCORM zip files, i.e. SCOs and assets, and create the games linking these LOs with specific events defined in the game template selected. The Web Game runs on the user browser. It includes the Game Template and the SGAME API. It communicates with the SGAME Platform using the SGAME API to request and show the corresponding LOs.

Finally we will use an example to explain the interactions that take place to show a LO when playing one of these

games. When a specific event occurs in the game, for example the player opens a chest, the Game Template calls the API to show one LO with the function *triggerLO* passing the number or name of the game event as a parameter. The SGAME API will call the platform that will deliver it together with its metadata, and it will be shown to the player. In the example a quiz with a couple of questions would appear and the player would have to answer them.

Whenever the player closes the LO the Game Template receives a LO *stats report* with information collected during the LO display time. In the example, the game would receive the number of questions that the student have correctly answered and will give the player 20 coins per each correct answer.

B. SGAME API

The SGAME API is the main contribution of this research. It is the one that enables the integration of SCORM packages into web games, i.e. the one that a game developer would have to use to create an adaptive web game or to edit his/her already developed games in order to convert them to a game template compliant with this model.

This API is quite simple. It only has two methods to communicate with the game template.

- An *Init* method with two options, firstly a *togglePause* function to stop the game when showing a LO and resume it when the LO is closed and secondly a *background image* for the display box where the LO is shown. This way the LO seems to be totally integrated in the game. Calling the Init method is not mandatory. It can be omitted if the game handles the pause/resume for itself. However, this way the LO will always appear with the default style, i.e. a white display box.
- A *triggerLO* method with the name or id of the game event. This is the main method in the API and will communicate with the SGAME Platform to request a LO and show it. Whenever the player closes the LO the game will receive two parameters, i.e. a true/false value called *success* indicating if the player consumed or passed the LO and a *report* with some statistics about what the player has done with the LO. The values will vary depending on the LO being an asset or a SCO:
 - If the LO is an asset the only information that we have about what the player did with the LO is the time spent (*tspent*). So the *success* parameter will follow a very simple algorithm, *true if tspent > tmin*. Being *tmin* a percentage of the “typical learning time” if present and a minimum time (that we initially set in 15 seconds) if not present. Anyway, the report contains this *tspent* for game developers who want to implement their own algorithm based on this data.
 - If the LO is a SCO it will automatically communicate with the Run Time Environment that is also included in the

SGAME API and it gathers much more information, i.e. completion rate, number of attempts, success indicators, etc. In this case the *success* parameter will be extracted from this information and the *report* will contain all the information gathered.

C. Conventions

A game developer that wants to follow this model would have to follow the next conventions for the created game:

- Include an index.html file that launches it.
- Include a file describing the events that can be linked to the LOs.
- Include the SGAME API.
- Implement a function to pause/resume the game when the LO is shown/closed.

V. SGAME PLATFORM: A CASE STUDY

Following this model we have developed a web platform that we have also called SGAME. In this section we will explain the main functionalities and an example of a generated game.

This platform is open to the public [20] and anyone can use it without registration being needed. In this version we have introduced three game templates. Two are existing games (Sokoban and Onslaught Arena) that are open source and we have modified to use the SGAME API, and another one called Natural Park that we have developed from scratch.

This platform is open source and the repository of the code can be found in GitHub [21]. Anyone can download it and it is open for contributions from the learning or game development communities. Furthermore, all the future works presented at the end of this paper will be also included in this repository.

A. Functionality

SGAME is an online e-Learning authoring tool to create educational web games by customizing the LOs and activities in them. Fig. 2 shows the main page of this platform.

The functionality is quite simple, the user has to choose a game template, some SCORM packages and click on the “create” button. A small form will be displayed to write a name for the created game and add an avatar to recognize it. After this the generated game is shown for the user to try it. The game will be always available in that URL so it can be easily shared. A “gallery” button to access a page with all the created games is also available.

In this screen the user can also upload his/her own SCORM packages. This way the user is not limited to use the existing ones.

Finally, there is a button where users will be able to add new game templates the same way as SCORM packages are added, although this functionality has not been implemented yet in the SGAME platform.



Fig. 2. SGAME platform

B. Example

Several examples of generated web games can be found in the SGAME platform gallery. We will explain one game example based on the game template Onslaught Arena.

This game is available online [22]. It is a medieval fantasy game where the user plays the role of a fighter that has to shoot different kind of monsters to get gold and weapons. Every time a player tries to get a new weapon (Fig. 3) a LO will be shown (Fig. 4). The player will only obtain the weapon by accomplishing or consuming the LO.

In Fig. 4 we can see that we have also included a semaphore that will show a red, yellow or green light as a feedback for the player. The status of this semaphore depends on the “typical learning time” parameter of the metadata or in the information gathered from the Run Time Environment API if the LO is a SCO. If he/she closes the LO before the green light is on he/she will not get the reward in the game, in this case the weapon. In this figure we see a quiz where the player has to drag and drop the weapon names to their corresponding pictures. After finishing this learning activity he/she will be able to continue playing.



Fig. 3. Game example

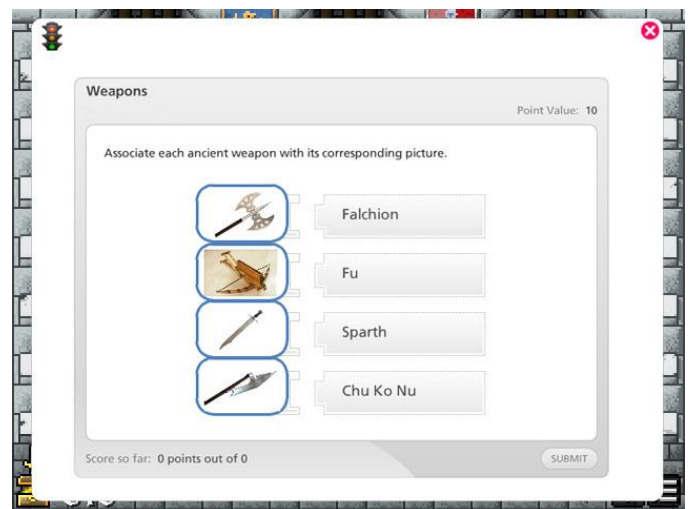


Fig. 4 Quiz shown in the game

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented SGAME, a model to integrate SCORM packages into web games. The main piece of this model is the SGAME API that games have to use to allow this integration. This API has been explained together with the model, architecture and interactions that occur when playing one of these generated games.

Following this model and to validate it we have developed an open source web platform (also called SGAME). We have also described it in this paper together with its functionality and an example of a generated game available online.

This model can bring closer the game developers community and the teaching community. The API is quite simple and can be easily integrated into existing web games or in new ones created from scratch. The possibilities that this model offers to the teachers are huge as they can customize games with their own learning content, this way adapting them for their students' age, language, subjects and areas of interest.

The immediate next step in this research consists of extending the model to other standards such as IMS Common Cartridge. In the platform the next work will be to allow a more flexible integration of the LOs with the events in the game, maybe with a second step in the creation process where the user links each LO with a specific event in the game. Together with this, it would be very interesting to allow the edition of the metadata, as we have seen that the information in them is very important. We could for example enable the user to specify the "typical learning time" of each LO inside the SCORM package in case it is not included.

Finally we would like to use the SGAME API in other contexts different than games, to allow the integration of SCORM packages in them, for example Android-iOS native applications.

ACKNOWLEDGMENT

We wish to acknowledge our gratitude and appreciation to all the GLOBAL excursion project partners, and each one of the project team members, for their contribution during the development of various ideas and concepts presented in this paper. We also would like to thank the SAAN project (TIN2010-19138) for funding this work.

REFERENCES

- [1] G. Denis and P. Jouvelot, "Motivation-driven educational game design," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '05*, 2005, pp. 462–465.
- [2] J. P. Gee, *What Video Games Have to Teach Us About Learning and Literacy. Second Edition: Revised and Updated Edition*. Palgrave Macmillan, 2007, p. 256.
- [3] G. Bekebrede, H. J. G. Warmelink, and I. S. Mayer, "Reviewing the need for gaming in education to accommodate the net generation," *Computers & Education*, vol. 57, no. 2, pp. 1521–1529, Sep. 2011.
- [4] "Advanced Distributed Learning (ADL), SCORM 2004 4th Edition." [Online]. Available: <http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th>.
- [5] T. M. Connolly, M. Stansfield, and T. Hainey, "An application of games-based learning within software engineering," *British Journal of Educational Technology*, vol. 38, no. 3, pp. 416–428, May 2007.

- [6] M. Papastergiou, "Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation," *Computers & Education*, vol. 52, no. 1, pp. 1–12, Jan. 2009.
- [7] K. D. Squire, "Video games in education," *Int. J. Intell. Games & Simulation*, vol. 2, no. 1, pp. 49–62, 2003.
- [8] A. Mitchell and C. Savill-Smith, "The use of computer and video games for learning: a review of the literature." Learning and Skills Development Agency, 02-Aug-2011.
- [9] S. De Freitas, "Learning in Immersive worlds. A review of game-based learning Prepared for the JISC e-Learning Programme (2006)." .
- [10] T. M. Connolly, E. A. Boyle, E. MacArthur, T. Hainey, and J. M. Boyle, "A systematic literature review of empirical evidence on computer games and serious games," *Computers & Education*, vol. 59, no. 2, pp. 661–686, Sep. 2012.
- [11] L. Anido, D. Burgos, M. Caeiro, J. Torrente, M. Fernandez, J. Gonzalez, M. Manso, M. Ortega, D. Rodriguez, and B. Fernandez-Manjon, "Game-Tel: An approach to multi-format and multi-device accessible engineering education," in *2011 Frontiers in Education Conference (FIE)*, 2011, pp. F1H–1–F1H–6.
- [12] R. Tornero-Santamarina, J. Torrente, P. Moreno-Ger, and B. Fernández-Manjón, "e-Training DS: An Authoring Tool for Integrating Portable Computer Science Games in e-Learning," *9th International Conference on WebBased Learning ICWL 2010*, vol. 21, no. 3, pp. 259–268, 2010.
- [13] "Stagecast." [Online]. Available: www.stagecast.com.
- [14] "ClickTeam." [Online]. Available: www.clickteam.com.
- [15] M. H. Overmars, "Game design in education," *UU-CS*, no. 2004–056, 2004.
- [16] H. D. Stolovitch and S. Thiagarajan, *Frame games*, vol. 24. Prentice Hall, 1980.
- [17] F. Mehm, S. Göbel, and R. Steinmetz, "Introducing Component-Based Templates into a Game Authoring Tool," in *5th European Conference on Games Based Learning*, 2011, pp. 395–403.
- [18] M. Frantzi, N. Moumoutzis, and S. Christodoulakis, "A Methodology for the Integration of SCORM with TV-Anytime for Achieving Interoperable Digital TV and e-Learning Applications," pp. 636–638, Aug. 2004.
- [19] I. L. T. S. Committee and others, "Draft standard for learning object metadata," *Accessed July*, vol. 14, p. 2002, 2002.
- [20] "SGAME web page." [Online]. Available: <http://sgame.dit.upm.es>.
- [21] "Github Repository." [Online]. Available: https://github.com/ebarra/scorm_into_games.
- [22] "Generated Game example." [Online]. Available: <http://sgame.dit.upm.es/game/1>.